An Implementation of a Barotropic Quasigeostrophic Model of Ocean Circulation on the MPP

C.E. Grosch and R. Fatoohi

Old Dominion University
Norfok, Virginia
and
Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, Virginia

Abstract

We discuss here the implementation on the MPP of a barotropic quasigeostrophic model of ocean circulation. The mathematical model, including scalings and boundary conditions is discussed. The numerical scheme, which uses compact differencing is also discussed. The implementation of this model on the MPP is then presented. Finally, some performance results are given and compared to results obtained using the VPS-32 and one processor of a CRAY-2.

Introduction

A central problem in the physics of ocean circulation is that of understanding the interactions of the spectrum of scales present in these motions. At the long wavelength end of the spectrum are the planetary scale motions, typified by the thermohaline gyre circulation, and the wind driven basin circulation. These dynamical features are reasonably quasisteady and are the background and, presumably, the forcing functions for the synoptic scale dynamics.

The synoptic scale motions are, roughly, on the scale of the radius of deformation and range from the meanders in the western boundary currents (one to two hundred kilometers), eddies (fifty kilometers), and fronts (ten kilometers or less) to the small scale eddies (one kilometer or less) which appear to be in the dissipation range. These motions would seem to be generated by instabilities in the large scale gyre circulation which results in the transfer of either kinetic or potential energy from the gyre to the synoptic scale motion. From both a theoretical and practical point of view, an understanding of the synoptic scale dynamics appears to be of major importance.

The fundamental force balances of the ocean are hydrostatic in the local vertical and geostrophic in the horizontal. All interesting ocean dynamics are the results of the small deviations from these balances. Numerical simulations of these dynamics in terms of the "primitive" variables, velocity, pressure, and density, are difficult because, among other problems, of the required high accuracy of the solutions in order to separate the time varying dynamics from the quasistatic geostrophic and hydrostatic balances. However, a complete prediction of the dynamics, including the density field, requires the use of the primitive system (Bryan, 1975). The primitive variable general circulation models are the only ones which can give a complete prediction of the dynamics, including, especially, the density field. Currently available computer resources are not sufficient to allow fine horizontal and vertical resolution models to be used for routine parametric experiments with these models.

The standard approximation used to effect the separation is the quasigeostrophic model. In this approximation the dynamical variables are the horizontal velocity components and the vertical component of the vorticity. The density field appears only in the depth varying Brunt-Vaisala frequency. This approximation is extraordinarily rich in phenomena,

containing, for example, wind driven basin circulation dynamics, Rossby waves, barotropic and baroclinic instabilities, as well as eddies. Within the quasigeostrophic approximation there is much physics whose effects should be studied in detail; in some cases the physics has been explicitly built into the models while in others it is parameterized. Examples of the former are topography and irregular basin geometry, wind forcing and surface thermal forcing, both of which should be time dependent; the outstanding example of the parameterized physics is that of the dissipation mechanisms, or more generally, the effects of sub-grid scale turbulent motions.

The fundamental physics of the wind driven large scale ocean circulation were deduced by Sverdrup (1947), Stommel (1948) and Munk (1950). Analytic models such as these are limited by the complexity of the nonlinear dynamics. Further progress in understanding required the use of numerical models. Some examples of the early use of numerical models are provided by the work of Bryan (1963), Veronis (1966) and Holland (1967). More recently Bretherton and Darweit (1975), Bryan (1975), Haidvogel, Robinson, and Schulman (1980), Robinson and Haidvogel (1980), Haidvogel (1983), Holland, Harrison, and Semtner (1983) and Miller, Robinson, and Haidvogel (1983), among others, have developed numerical models of quasigeostrophic flow and used them to study ocean dynamics.

In their most general form these models require very large amounts of computation time on the fastest computers available. Despite the fact that the most powerful existing vector processors can perform at peak rates of hundred of MFLOPS, they are inadequate for many applications. In part this inadequacy is because it is generally rather difficult to fully use the vector capabilities of these supercomputers. In practice, average processing rates for many codes are in the range of 10 to 20 MFLOPS (see, for example, Dongarra, 1984). An alternative way of achieveing greater processing power is to use computers consisting of multiple (hundreds or thousands) processing elements; each processing element has only a modest processing power and storage. However, a complete multiple processing system can have a very large processing capability. If these multiprocessor computers can be used effectively, very large gains in overall processing power are possible.

In this paper we report on the implemenation of a quasigeostrophic potential vorticity model on the Massively Parallel Processor (MPP). In section 2 we give the scalings, basic assumptions, equations, and boundary conditions. The numerical scheme is outlined in section 3. Section 4 contains a discussion of the implementation of this model on the MPP. Finally, in section 5 the preliminary results of this research are given and some conclusions are drawn.

Equations

The scaling of the variables and the derivation of the quasigeostrophic equations follows the development of Pedlosky (1979). The x axis is positive eastward and the y axis is positive northward. The coriolis parameter is approximated by the beta plane approximation.

$$f = 2\Omega \sin\phi \approx f_0 + \beta_0 \tilde{y}, \qquad (2.1a)$$

with

$$f_0 = 2\Omega \sin\phi_0, \qquad (2.1b)$$

$$\beta_0 = (2\Omega / r_0) \cos\phi_0, \qquad (2.1c)$$

where Ω is the rotational frequency of the earth, r_0 is the radius of the earth, and $\tilde{y}$ is the dimensional distance northward of the latitude ϕ_0 (see Figure 1). A lateral length scale, L, and a vertical length scale, D, are used. It is assumed that D/L is much less than unity. The horizontal velocity components, u to the East and v to the North are scaled by U. The time is scaled by $U/\beta L$ where

$$\beta = \beta_0 L^2 / U. \qquad (2.2)$$

Expanding the solution in powers of the Rossby number(U/foL), the equations of the quasigeostrophic model for the depth averaged horizontal velocity $\vec{u}$ =(u,v) and vertical component of the vorticity, ζ are:

$$u_x + v_y = 0$$
$$v_x - u_y = \zeta$$

$$\zeta_t + S(u\zeta_x + v\zeta_y) + \gamma\zeta + v =$$
$$\hat{k}\cdot(\nabla\times\vec{\tau}) + SRe^{-1}(\zeta_{xx} + \zeta_{yy})$$

$$S = \beta^{-1}, \quad \gamma \text{ is} \qquad (2.5)$$

dimensionless bottom friction coefficient and $\vec{\tau}$ is the surface wind stress. The wind stress is dimensionless and its magnitude, τ_0, has been scaled to one by taking

$$U = \tau_0/\rho D\beta_0 L. \qquad (2.6)$$

Finally the Reynolds number, Re, is defined by

$$Re = UL/A, \qquad (2.7)$$

with A a horizontal eddy viscosity. In this derivation it was assumed that the density was constant, that the bottom was flat and that the wind stress, $\vec{\tau}$, was applied to a rigid lid on the surface.

If these equations are applied to the flow in a closed basin the boundary conditions are quite simple, $\vec{u}$=0 on the boundary. This implies that the vorticity on the boundary is equal to the normal derivative of the velocity on the boundary.

For application to flow in basins which are partly or completely open, inflow and outflow boundary conditions must be imposed. This is a subject of continuing research and will not be discussed further.

It should be noted that this formulation of the barotropic quasigestrophic equations is in terms of velocity and vorticity. This is in contrast to most other formulations in which a stream function is introduced to satisfy equation (2.3). Among the few to use the velocity-vorticity formulation in numerical calculations are Dennis, Ingram, and Cook (1979), Fasel (1980), and Gatski, Grosch, and Rose (1982). The velocity, vorticity formulation of the Navier-Stokes equations has certain advantages over more conventional formulations, particularly with respect to setting boundary conditions. For example, it is quite easy to incorporate the generation of vorticity at solid boundaries, and at outflow boundaries to use a vorticity flux condition (Halpern, 1985) to ensure that no vorticity is artificially reflected back into the computational domain.

Numerical Scheme

The numerical method used is based on compact differencing schemes. These schemes require the use of only the values of the dependent variables in and on the boundaries of a single computational cell. Compact difference methods have been discussed by Keller (1974). Malik, Chuang, and Hussaini (1982) used a fourth-order, compact, difference scheme to solve the compressible linear stability equations, and Gatski, Grosch, and Rose (1982) applied second-order compact schemes to the numerical solution of the incomprehensible, two-dimensional, time-dependent Navier-Stokes equations. This code was later extended to three dimensional time-dependent flows and used to study some complex flow fields (Gatski and Grosch, 1985a,b). The original ocean circulation code was developed as a serial algorithm by Spence and Grosch (1985).

In order to apply the compact differencing scheme we must write the vorticity equation as a first order system. Define

$$\phi = \zeta_x \qquad (3.1a)$$

$$\psi = \zeta_y \qquad (3.1b)$$

then (2.5) becomes

$$\zeta_t + S(u\zeta_x + v\zeta_y) + \gamma\zeta + v = \qquad (3.2)$$
$$\hat{k}\cdot(\nabla\times\vec{\tau}) + SRe^{-1}(\phi_x + \psi_y).$$

The velocity and vorticity at time levels $n\Delta t$ and $(n+\frac{1}{2})\Delta t$ are located on the boundaries of and in a cell as shown in Figure 2.

The centered difference and average operators are defined on a cell by

$$\delta_x U^n \equiv (U^n_{i+1/2,j} - U^n_{i-1/2,j})/\Delta x, \quad (3.3)$$

$$\mu_x U^n \equiv (U^n_{i+1/2,j} + U^n_{i-1/2,j})/2. \quad (3.4)$$

With these definitions the, equations (2.3), (2.4) replaced by

$$\delta_x U^n + \delta_y V^n = 0, \quad (3.5)$$

$$\delta_x V^n - \delta_y U^n = \zeta^{n-1/2}, \quad (3.6)$$

$$\mu_x U^n - \mu_y U^n = 0, \quad (3.7)$$

$$\mu_x V^n - \mu_y V^n = 0 \quad (3.8)$$

The vorticity transport system, equations (3.1a), (3.1b), and (3.2) are replaced by

$$[\delta_t + S(\mu_x U^n)\delta_x + S(\mu_y V^n)\delta_y + \gamma]\zeta^n = F^n - \mu_x\mu_y V^n + S\,Re^{-1}[\delta_x\phi^n + \delta_y\psi^n], \quad (3.9)$$

$$\delta_x\zeta^n = (\mu_x - \tfrac{1}{2}\Delta x\, q(\theta_x)\delta_x)\phi^n, \quad (3.10)$$

$$\delta_y\zeta^n = (\mu_y - \tfrac{1}{2}\Delta x\, q(\theta_y)\delta_y)\psi^n, \quad (3.11)$$

$$\mu_t\zeta^n = \mu_x\zeta^n = \mu_y\zeta^n, \quad (3.12)$$

and

$$q(\theta) \equiv \coth\theta - 1/\theta, \quad (3.13)$$

with

$$\theta_x = S U^n Re\,\Delta x/2, \quad (3.14)$$

$$\theta_y = S V^n Re\,\Delta y/2, \quad (3.15)$$

the cell Reynolds numbers. Note that $F^n_{i,j}$ is the cell averaged curl of the wind stress at time $n\Delta t$.

The boundary condition for the velocity equations (3.5) to (3.8) are that the normal component is zero on a solid boundary. This produces a nonzero tangential component on the boundary, which in turn gives the boundary condition for the vorticity equations (3.9) through (3.15); enough vorticity is produced at the wall so that the tangential velocity is brought to zero.

Implementation on the MPP

In order to map the data onto the MPP, it is convenient to introduce auxiliary or box variables to represent the velocity field. Referring to figure 2, we introduce the box variables P and Q at the corners of the cells. These are defined so that the average of two adjacent P's is equal to the U on the included side and similarly for the Q's and V. We then map this cell onto the array so that each corner of the cell "corresponds" to a processor and holds a P and a Q value. Again referring to Figure 2, if one considers the processor at the upper left hand corner of the cell the value of the vorticity at the center of the cell and on the left hand and bottom sides, as well as the corresponding values of ϕ and ψ, are stored in that processor. Thus, the array with 128 by 128 processors is mapped onto a 127 by 127 array of cells.

It is easy to see that, expressing U and V in terms of the P's and Q's, equations (3.7) and (3.8) are satisfied identically. One must then solve (3.5) and (3.6) with given and the boundary condition that the normal component of the velocity is zero on the boundary of the domain. This is done using a cell relaxation scheme in which all the values of P and Q on the corners of a cell are updated simultaneously, for details see Gatski, et al. (1982). It is obvious that the values of P and Q at the corner of any cell are updated four times in a sequential sweep across the domain. Thus, this method is a four "color" scheme. It is quite simple to update each of the "colors" in parallel and four updates are required for a complete update of all the points in the domain. The relaxation scheme is equivalent to an SOR method.

Once a solution for the P's and Q's has been found, the next step in the solution procedure is to calculate the boundary values of the vorticity. This is done by computing the vorticity required to be added at the boundary so that the tangential component of the velocity is reduced to zero. This is done using equation (2.4).

Once the values of the vorticity on the boundary are known, one can proceed to solve the vorticity equations (3.9) through (3.15). Using equation (3.12) the vorticity at time level n+1/2 can be eliminated from (3.9). This, together with the other equations, gives an implicit system for the vorticity at time . This system is solved using an ADI method which requires the solution of tridiagonal systems of equations in, alternately, all rows and all columns. This is done in parallel on the array using the cyclic elimination algorithm. Once the vorticity at time is known as well as at (n-1/2) , equation (3.12) can be used to compute the vorticity at time (n+1/2) . Finally, various norms, the total energy and so on are computed. The mathematical details of this algorithm are described in Gatski et al. (1982).

Results and Discussion

The algorithm described above has been coded in MPP Pascal and run on the MPP. The code is moderate length; between 1300 and 1400 lines. This can be compared with the serial FORTRAN version which is about 2000 lines long. The code for the four color relaxation is simple and highly structured. In contrast the code for the ADI vorticity solver is much more complex and less structured, it also uses about 20 temporary parallel matrices. There are procedures to calculate an approximation to the q function, equation (3.13) and to solve tridiagonal equations over the rows and columns. Despite this complexity, the time used by the ADI solver is only equal to ten or so iterations of the velocity solver. When the wind stress is varying in time or there are Rossby waves propagating in the basin the velocity solver requires a few hundred iterations per time step and, thus, uses most of the computational time. Therefore the performance of the parallel relaxation routing largely determines that of the entire code.

Processing rates, usually expressed in megaflops, are one measure of the performance of an algorithm on a computer. Comparison of the processing rates of the same algorithm on different computers can be misleading because of the differences in the basic cycle time of the machines and because the same algorithms may require different numbers and kinds of operations when optimized for different architectures. Neverhtless, cross comparisons are useful when a comparison is also made with the maximum theoretical processing rate of the computers. This kind of comparison shows how well, or poorly, a particular algorithm has been adapted to each of the architectures. Such a comparison is shown in Table 1.

The relaxation algorithm, in addition to being programmed for the MPP, has been programmed in CDC Vector FORTRAN for the VPS-32, and in FORTRAN for the CRAY-2. The programs were run on each of these machines and timings were obtained. These, together with the operation counts from the codes permitted the arithmetic processing rates to be determined. These together with the theoretical maximum rates are given in Table 1 for a 128 x 128 grid point problem.

The scalar rates shown in this table were obtained when standard FORTRAN codes embodying a serial version of the algorithm were run through a vectorizing compiler. None of the code was vectorzied. The vector performance on the VPS-32 and the CRAY-2 was obtained after the FORTRAN codes were rewritten, explicitly using the four color structure of the algorithm. In addition, the explicit Vector FORTRAN was used for the VPS-32 code and the CRAY compiler had to be told to ignore apparent vector dependencies.

This relaxation algorithm mapped well onto the MPP. This is because it can be set up as nearly all matrix operations. There are relatively few shift operations and these are all to nearest neighbors. There are no vector operations and only two scaler operations per iteration.

In contrast this algorithm does not vectorize as well. Because of the color structure, the longest vectors are one-half of the maximum dimension; in this case 64. This contrasts with N-1/2 value of 110 for the VPS-32. The structure of the algorithm also requires that the data be accessed with a stride of 2 and this requires the extensive use of COMPRESS and MERGE functions. This adds a considerable

overhead. Increasing the problem size will improve the vector length problem but will not remove the overhead associated with the slow COMPRESS and MERGE functions.

On the CRAY-2 the stride of 2 is a source of major difficulty. All of the inner loops of the rewritten code vectorized, but the major problem was loading the data from the main memory to the local memory or the registers. Here the stride of 2 caused bank conflicts.

A perhaps more realistic way of measuring performance is to consider the time required to solve a given problem. Table 2 contains a listing of the measured time required to complete one iteration of the 128 x 128 problem on the MPP, the VPS-32, and the CRAY-2. The execution time for the CRAY is about 30% less than that of the MPP, while that of the VPS-32 is nearly 2.5 times greater than that of the MPP. We also note that we do more operations on the MPP than on the CRAY.

A crucial question in using the MPP is whether or not a problem will fit on the array or is oversize. Each of the major procedures in this code requires 20 to 25 arrays of floating point numbers. Many of these are temporary arrrays, but these together with other storage require that we move various arrays in the out of the staging memory even when we have a 128 x 128 problem. Therefore, we must partition these oversize problems and use the staging memory as a backup. We have modified the algorithms so as to be able to handle oversize algorithms, the details will appear elsewhere. A total of seven data swaps between the stager and the array are required for each sheet of data. The swapping adds 11.5 msec to the time for each iteration; thus one iteration on a sheet requires 25.1 msec. Some results for oversize problems, both measured and extrapolated, are given in table 2.

In summary we have found that it is possible to adapt a barotropic quasigeostrophic potential vorticity code to the MPP. We have been able to use the MPP's considerable processing power to solve the computationally intensive flow problems of ocean circulation. For problems which fit on the array, one iteration requires about 14 msec; this is about 30% more than that required on one processor a CRAY-2, and only about 40% of the time required for the same calcualtion on a VPS-32. For oversize problems, 2, 3, ..., 10 times the array size there is an overhead for transferring the data to and from the staging memory. This reduces the efficiency of the MPP, but not drastically. It is still competitive with the other supercomputers.

The problems which we have encountered in using the MPP have not been major. With regards to software, MPP Pascal as currently implemented, does not permit operations on vectors. This is awkward when dealing with boundary conditions. The major hardware problem is that the PE memory is only marginally large enough for our applications. We really need 4K to 8K bits per PE for the barotropic code and 32K bits per PE for a baroclinic model. Finally, input/output is somewhat clumsy.

Despite these problems, we believe that the MPP is a useful tool for running ocean circulation models and we will continue using it to study the effects of temporal and spatial variation in the wind field.

REFERENCES

Bretherton, F.P., and M. Karweit, "Mid-Ocean Mesoscale Modeling," In: *Numerical Models of Ocean Circulation*, National Academy of Sciences, 94, 1975.

Bryan, K., "Three-Dimensional Numerical Models of the Ocean Circulation," In: *Numerical Models of Ocean Circulation,* National Academy of Sciences, 94, 1975.

Dennis, S.C.R., D.B. Ingram, and R.N. Cook, "Finite Difference Methods for Calculating Steady Incompressible Flows in Three Dimensions," *Journal of Computational Physics,* **33**, 325, 1979.

Dongarra, J.J., "Performance of Various Computers Using Standard Linear Equations Software in a FORTRAN Environment," Argonne National Laboratory Technical Memorandum 23, 1984.

Fasel, H.F., "Numerical Solution of the Complete Navier-Stokes Equations for the Simulation of Unsteady Flows," Lecture Notes in Mathematics, Number 771, Springer-Verlag, New York/Berlin, 1980.

Gatski, T.B, and C.E. Grosch, "A Numerical Study of the Two- and Three-Dimensional Unsteady Navier-Stokes Equations in Velocity- Vorticity Variables Using Compact Schemes," *Ninth International Conference on Numerical Methods Fluid Dynamics,* Springer-Verlag, 235, 1985a.

Gatski, T.B., and C.E. Grosch, "Embedded Cavity Drag in Steady Laminar Flow, " *AIAA Journal,* **23**, 1028, 1985b.

Gatski, T.B., C.E. Grosch, and M.E. Rose, "A Numerical Study of the Two-Dimensional Navier-Stokes Equations in Vorticity- Velocity Variables," *Journal of Computational Physics,* **48**, 1, 1982.

Haidvogel, D.B., A.R. Robinson, and E.E. Schulman, "The Accuracy, Efficiency, and Stability of Three Numerical Models with Applications to Open Ocean Problems," *Journal of Computational Physics,* **34**, 1, 1980.

Haidvogel, D.B., "Periodic and Regional Models," In: *Eddies in Marine Science,* Ed. A.R. Robinson, Springer-Verlag, 404, 1983.

Halpern, L., "Artificial Boundary Conditions for the Linear Advection Diffusion Equation," CMAP-Ecole Polytechnique, Paris Report #118, 1985.

Holland, W.R., D.E. Harrison, and A.J. Semtner, Jr., "Eddy-Resolving Numerical Models of Large-Scale Ocean Circulation," In: *Eddies in Marine Science*, Ed. A.R. Robinsons, 379, 1983.

Keller, H.B., "Accurate Difference Methods for Nonlinear Two- Point Boundary Value Problems, " *SIAM Journal on Numerical Analysis,* **11**, 305, 1974.

Malik, M.R., S. Chuang, and M.Y. Hussaini, "Accurate Numerical Solution of Compressible Linear Stability Equations," *Journal of Applied Mathematical Physics,* (ZAMP), **33**, 189, 1982.

Miller, R.N., A.R. Robinson, and D.B. Haidvogel,, "A Baroclinic Quasigeostrophic Open Ocean Model," *Journal of Computational Physics,* **50**, 38, 1983.

Munk, W.H., "On the Wind-Driven Ocean Circulation, " *Journal of Meteorology,* **7**, 79, 1950.

Pedlosky, J., *Geophysical Fluid Dynamics,* Springer-Verlag, Berlin, 1979.

Robinson, A.R., and D.B. Haidvogel, "Dynamic Forecast Experiments with a Barotropic Open Ocean Model," *J. Phys. Ocean,* **10**, 1909, 1980.

Robinson, A.R., "Overview and Summary of Eddy Science," In: *Eddies in Marine Science,* Ed. A.R. Robinson, Springer-Verlag,3, 1983.

Stommel, H., "The Westward Intensification of Wind-Driven Ocean Currents," *Trans. Amer. Geo. Union,* **29**, 202, 1948.

Sverdrup, H.V., "Wind Driven Currents in a Baroclinic Ocean; With Application to the Equatorial Currents of the Eastern Pacific," *Proceedings of the National Academy of Sciences,* **33**, 318, 1947.

	MPP	*VPS-32*	*Cray-2* (*)
Theoretical Maximum Rate	210	200	488
Measured Vector or Array Rate	175	48	102
Measured Scalar Rate	--	7	30

(* Using one processor)

Table 1. Theoretical and measured processing rates, in Megaflops, using the relaxation algorithm on a 128 x 128 grid.

	MPP	*VPS-32*	*Cray-2*
Problem Size			
128 x 128	13.6	33.4	10.4
128 x 255	50.2	47.9	20.7
128 x 509*	100.4	73.6	41.4
128 x 1017*	200.8	124.0	82.6

(* Computed by extrapolation)

Table 2. Execution time, in milliseconds, for one iteration. One processor the CRAY-2 was used.

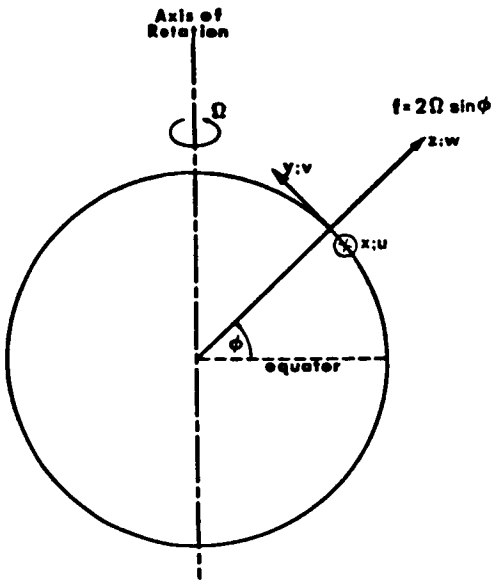

Figure 1. Coordinate System

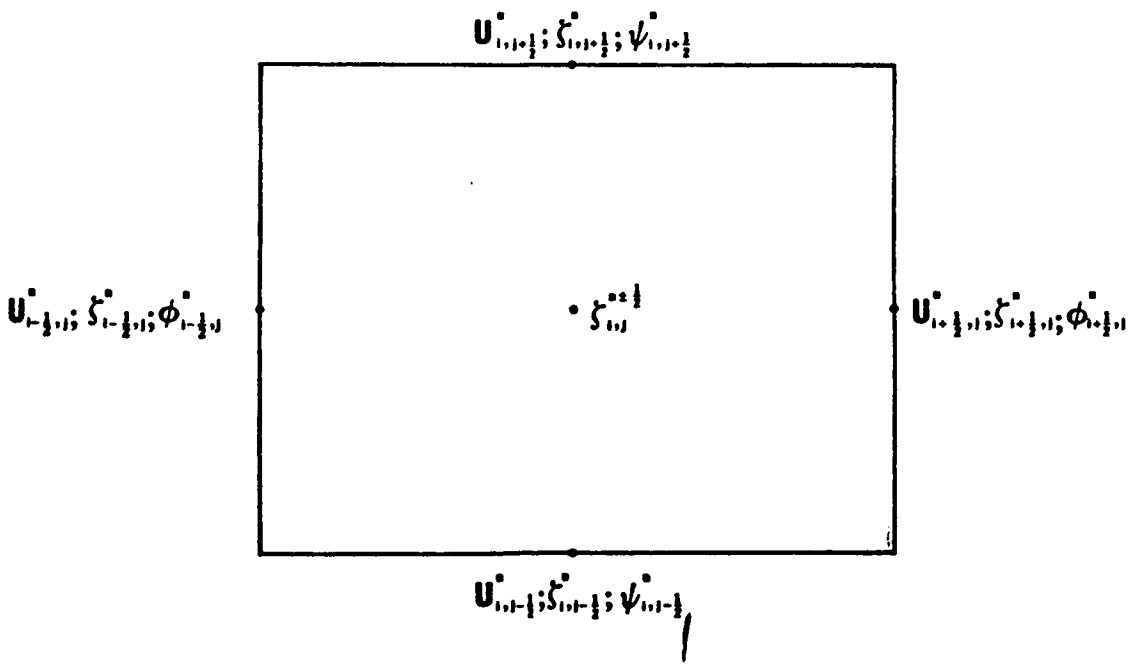

Figure 2. Mesh cell and location of variables.